Based on slides by Harsha V. Madhyastha

EECS 482 Introduction to Operating Systems

Spring/Summer 2020

Lecture 20: File systems recap

Nicole Hamilton
https://web.eecs.umich.edu/~nham/
nham@umich.edu

Agenda

1. Recap of storage devices.
2. Project 4 preview.

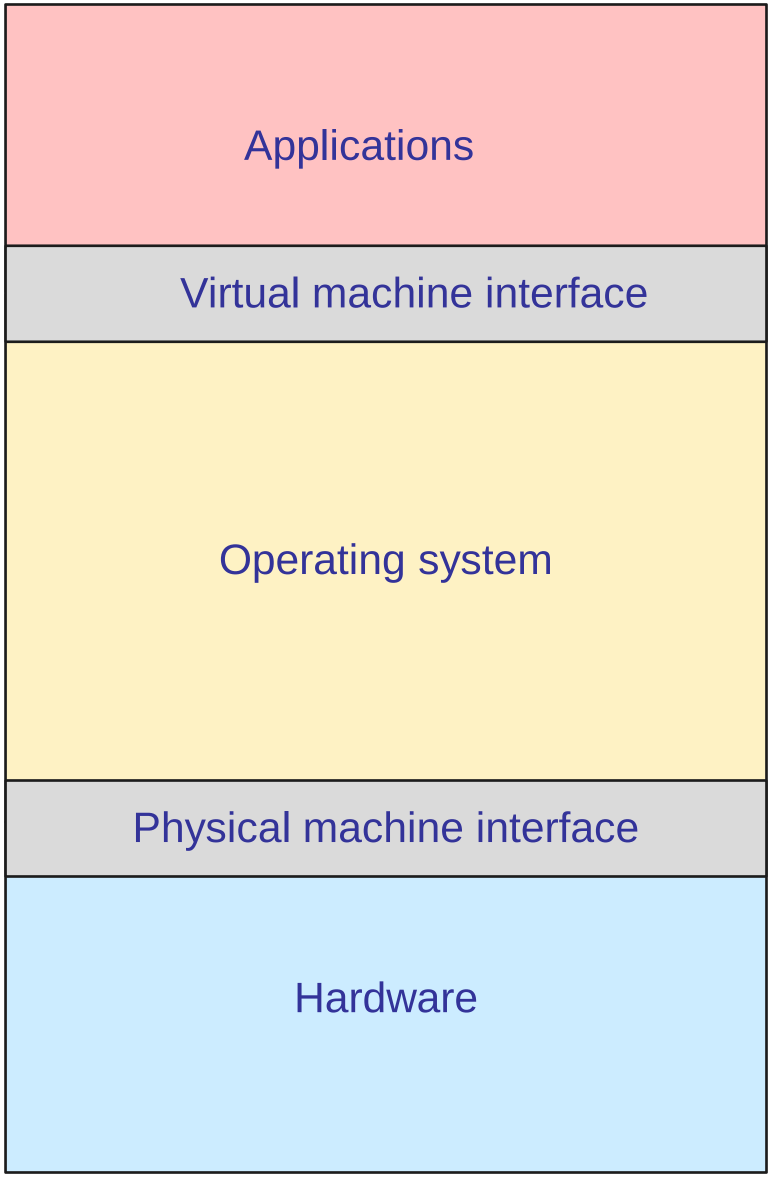
Agenda

1. Recap of storage devices.
2. Project 4 preview.

Dealing with heterogeneity

Many different types of disks and other devices and lots of different interfaces, e.g., ESDI, USB, SCSI, SATA, Fiber channel, m.2.

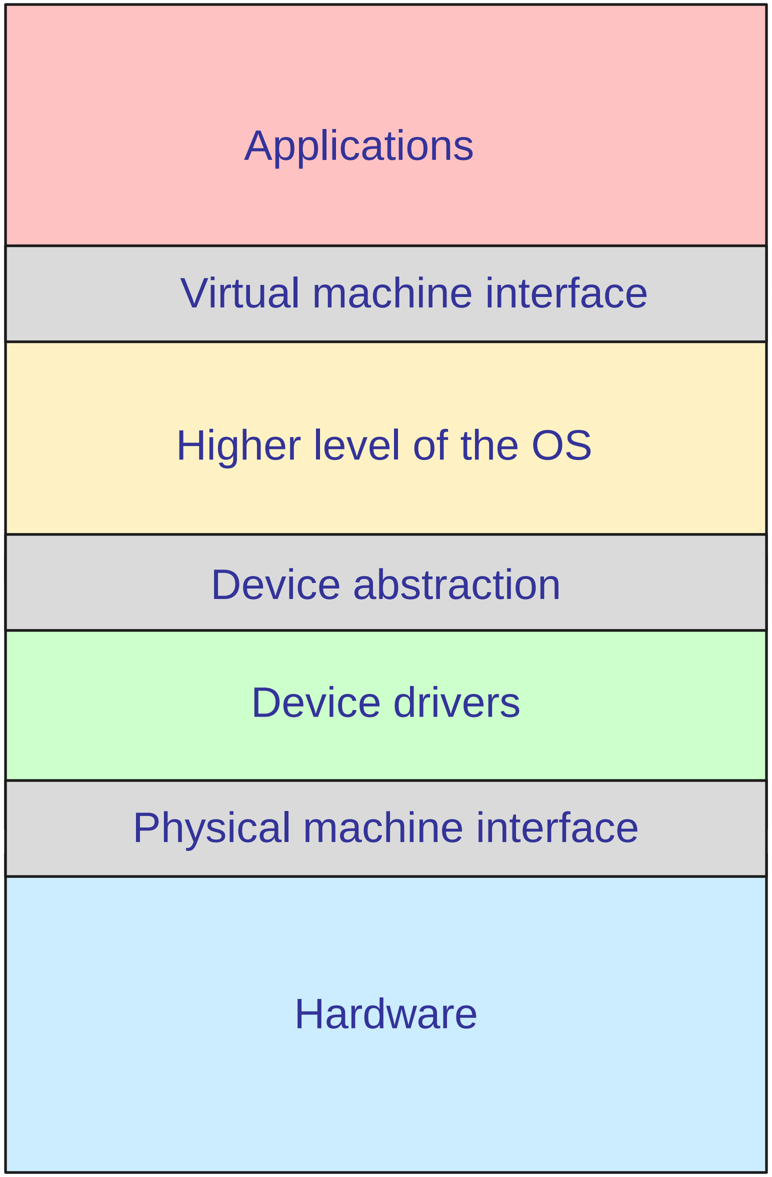
Need a way of managing this diversity.



Dealing with heterogeneity

Solution is to add a device driver abstraction inside the operating system to hide the differences between similar classes of devices.

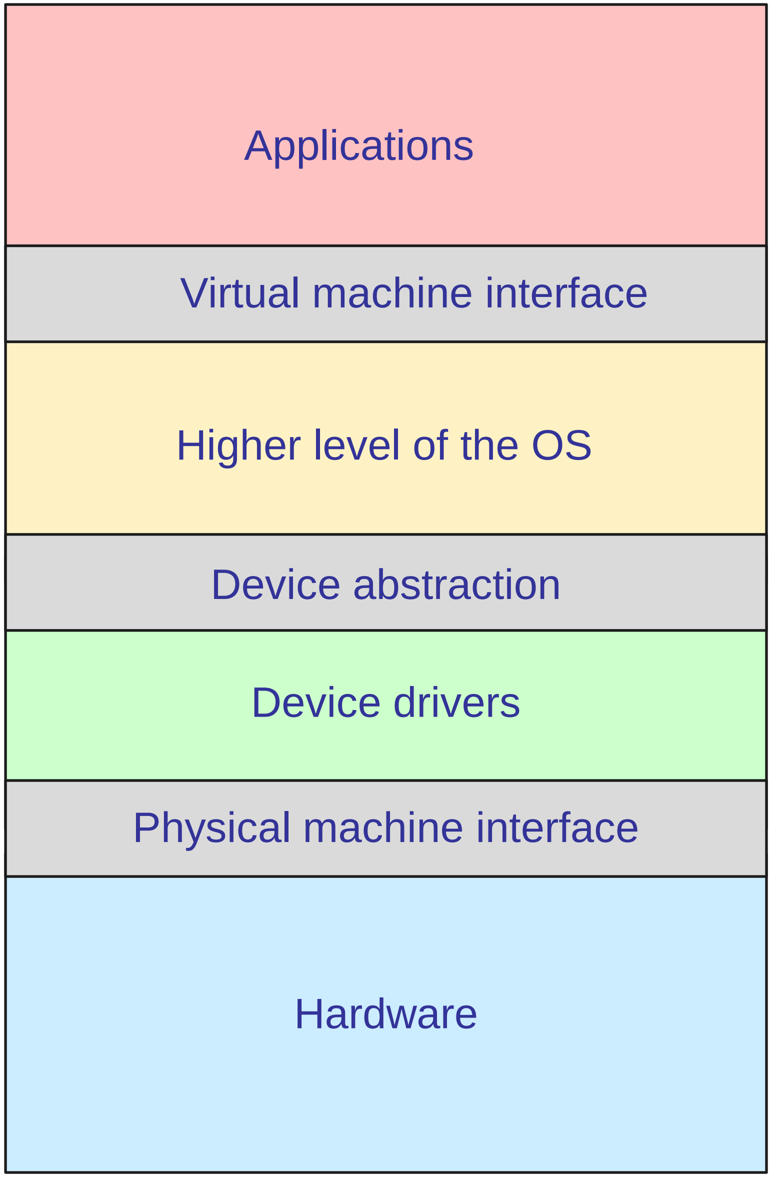
Device drivers create an abstraction of a disk as array of disk blocks.

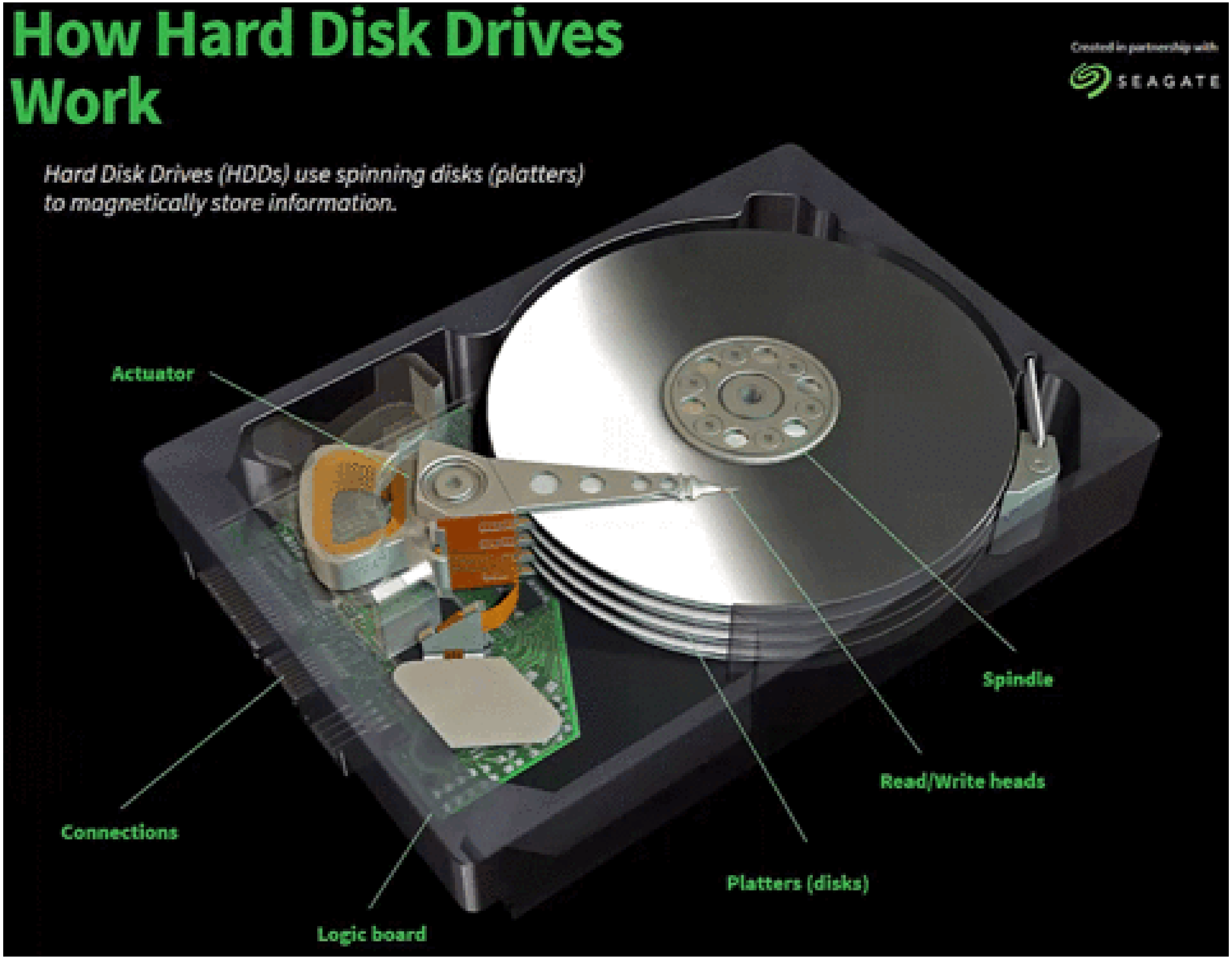


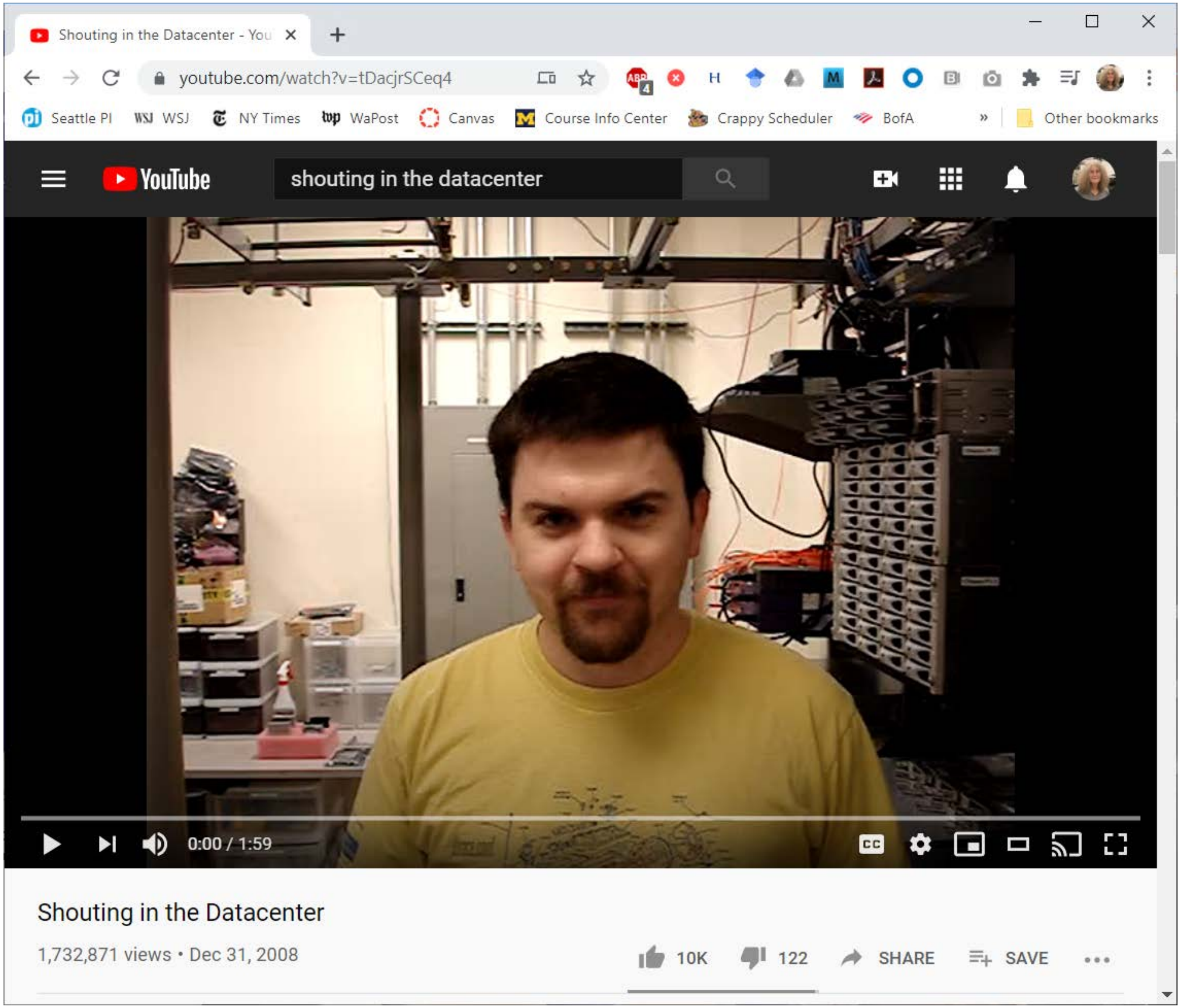
Dealing with heterogeneity

Device drivers are usually supplied by the device manufacturers.

Because they run as a trusted part of the kernel, in the past, they’ve been a major reason for Windows crashes.




How Hard Disk Drives Work
Created in partnership with
SEAGATE
Hard Disk Drives (HDDs) use spinning disks (platters) to magnetically store information.
Actuator
Spindle
Read/Write heads
Platters (disks)
Connections
Logic board



https://www.youtube.com/watch?v=tDacjrSCeq4

Hard Disk Performance

What does disk performance depend upon?

Queue	Wait for the disk to be free.
Positioning	Move the disk arm to the correct cylinder and rotate to the right sector.
Access	Transfer data from/to disk.

For given load, performance depends on

- Positioning overhead, called seek time, ~1 to 10 ms.
- Transfer time, ~100 MBps.

Disks Heterogeneity

- Seagate Barracuda 3.5" (workstation)
  - capacity: 250 - 750 GB
  - rotational speed: 7,200 RPM
  - sequential read performance: 78 MB/s (outer) - 44 MB/s (inner)
  - seek time (average): 8.1 ms
- Seagate Cheetah 3.5" (server)
  - capacity: 73 - 300 GB
  - rotational speed: 15,000 RPM
  - sequential read performance: 135 MB/s (outer) - 82 MB/s (inner)
  - seek time (average): 3.8 ms
- Seagate Savvio 2.5" (smaller form factor)
  - capacity: 73 GB
  - rotational speed: 10,000 RPM
  - sequential read performance: 62 MB/s (outer) - 42 MB/s (inner)
  - seek time (average): 4.3 ms

Optimizing I/O performance

To increase performance of slow I/O devices:

- Avoid doing I/O (Disks are *slooooow!*)
- Reduce overhead (minimize positioning time)
- Amortize overhead over larger requests

Efficiency = transfer time / (seek time + transfer time)

- Rule of thumb: Achieve at least 50% efficiency
- Example: 5 ms average seek time and 100MBps transfer rate → Read at least 500KB

Optimizing I/O performance

Two important approaches:

1. Improve the order in which requests are served.
2. Keep data that goes together close together.

Disk scheduling

Reduce overhead by reordering requests.

Can be implemented in OS or hardware.

Tradeoffs?

The hardware knows more about the device itself, bad blocks, error handling, what's really likely to be faster, and can offload work from the OS.

The OS knows more about the application needs.

FCFS

Pick 1 of n requests in queue:

- Example: 98, 183, 37, 122, 14, 124, 65, 67
- Start track is 53

FCFS (first come, first served)

- 98, 183, 37, 122, 14, 124, 65, 67
- Total head movement: 640 tracks

SSTF (STCF)

Pick 1 of n requests in queue:

- Example: 98, 183, 37, 122, 14, 124, 65, 67
- Start track is 53

SSTF (shortest seek time first)

- 65, 67, 37, 14, 98, 122, 124, 183
- Total head movement: 236 tracks

Any drawbacks?

- Potential starvation. Some tracks may never get served.

SCAN (Elevator)

Pick 1 of n requests in queue:

- Example: 98, 183, 37, 122, 14, 124, 65, 67
- Start track is 53

SCAN (like windshield wipers)

- 37, 14, 65, 67, 98, 122, 124, 183
- Total head movement: 208 tracks

Drawbacks and fix?

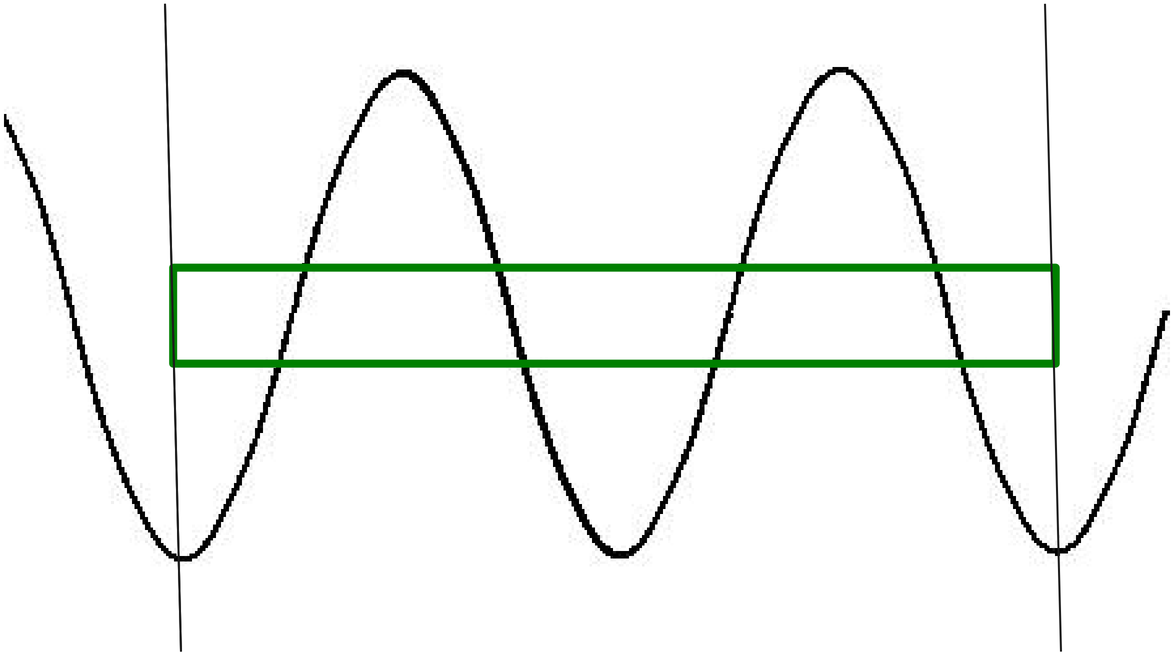
- Blocks in the middle served more often than at ends.
- Circular SCAN (C-SCAN): Serve requests only in one direction

SCAN

Head position

Consider two cycles.

Over 2 cycles, the head enters the middle 4 times.



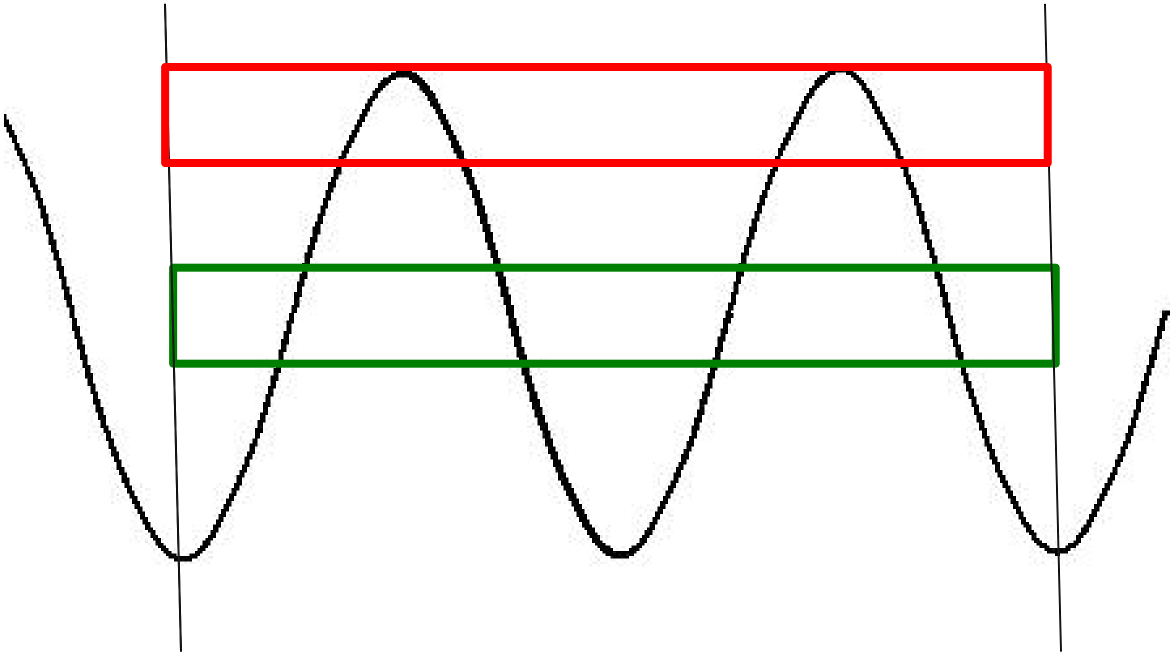
SCAN

Head position

Consider two cycles.

Over 2 cycles, the head enters the middle 4 times.

But it enters a given peak only twice.



SCAN vs. SSTF

- SCAN typically has better throughput
  - Minimizing total head movement
- SSTF may have better response time
  - Servicing fastest request first
  - But, poor throughput can cause longer queue waits
- Disk scheduling: Queueing vs. positioning
- Does CPU scheduling affect throughput?

Anticipatory scheduling

Consider two processes with disk locality

- P1: read 1, compute, read 2, compute, read 3, ...
- P2: read 1001, compute, read 1002, compute, ...

What behavior will SSTF give?

- 1, 1001, 2, 1002, 3, ... (suboptimal)

Key idea: wait for a bit for new request

- 1, 2, 3, (timer expires), 1001, 1002
- Can improve throughput **and** response time

Optimizing data layout

Keep related items together on disk, e.g., on the same track or same cylinder.

What items will be accessed together?

Can guess based on general usage patterns.

- Blocks in same file often accessed together.
- Files in same directory often accessed together.
- Files often accessed with their directory.

Can guess based on past accesses of data.

- Learn patterns and reorganize data on disk.

Flash (solid state disks)

Optimizations depend on specifics of a device.

Flash differs from magnetic disk.

1. Better random read performance.
2. Lower positioning overhead.
3. Much faster transfer rates (m.2).
4. Starting to yield more read parallelism.
5. Lower power.
6. Better shock resistance.
7. But they experience *wearout*, a limited number of times a cell can be rewritten.

OS hides physical characteristics of device from applications.

Optimizing for Flash

Move data blocks to do wear leveling.

Write data in big blocks.

Asynchronously erase blocks.

Prefer to read data rather than write.

Agenda

1. Recap of storage devices.
2. Project 4 preview.

Project 4: due August 17

Secure, multi-threaded network file server

- Network programming, file systems, client-server, threads/concurrency, even a little security.
- Experience writing significant concurrent program.

Good news: concepts simpler than projects 2 and 3.

Bad news: Perhaps 3x as much code as project 3.

Make sure to try out Friday's lab questions.